RJags

Se lancer dans les statistiques Bayésiennes avec R

Qui suis-je?

Sonia Eynard

Postdoctorat à l'INRA, unité GenPhySE

Projet : Recherche de marqueurs génétiques liés à la résistance au Varroa dans la population d'abeilles mellifère française

Formation : Master en conservation de la biodiversité et génétique des populations
Thèse génétique animale, impact de la sélection sur la diversité génétique chez le bétail

Qui suis-je?

Sonia Eynard

Po

Pro

Fo

Pas une pro des statistiques Bayésiennes ni de RJags

Essai d'utilisation pour analyser des données un peu plus complexes

Apprentissage en cours

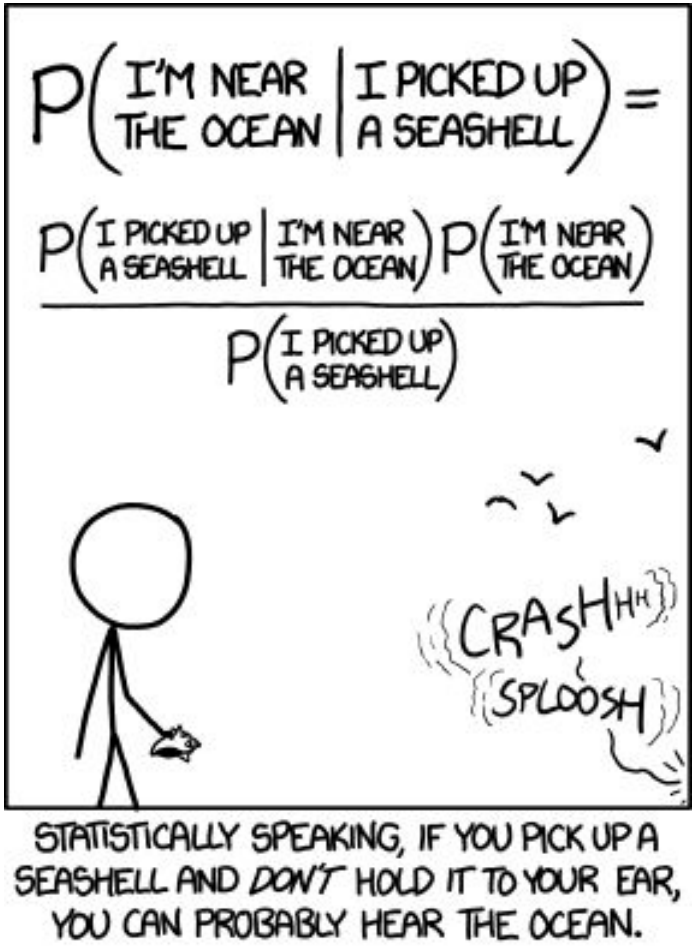
Thèse génétique animale, impact de la sélection sur la diversité génétique chez le bétail

Les statistiques Bayésiennes

(en 1 diapo)

H=hypothèse et D=données

$$p(H \mid D) = p(H)\, p(D \mid H) / p(D)$$



http://imgs.xkcd.com/comics/seashell.png

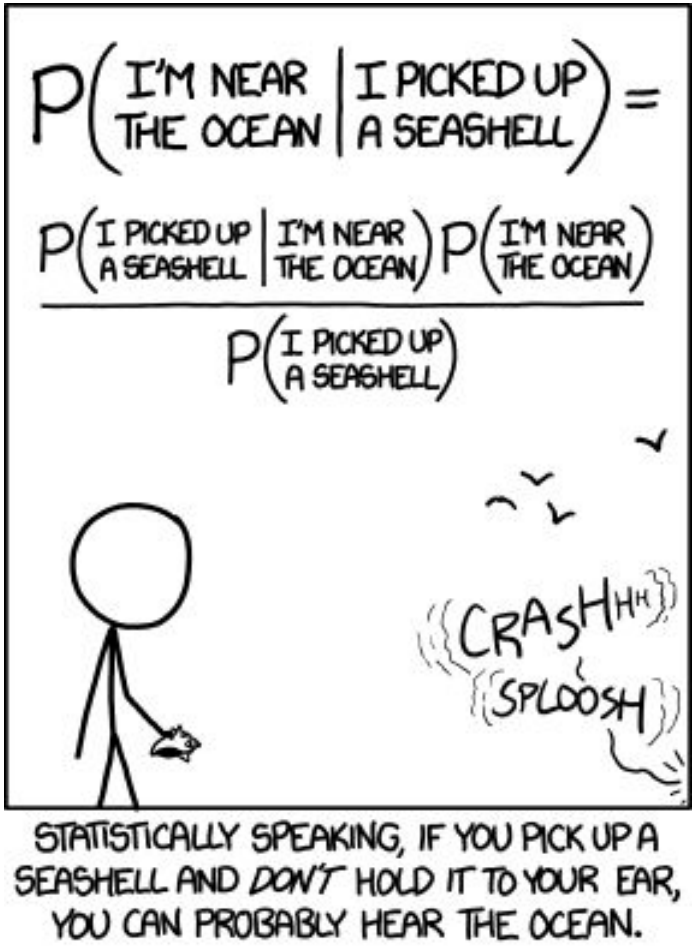
Les statistiques Bayésiennes

(en 1 diapo)

H=hypothèse et D=données

probabilité de l'hypothèse H connaissant les données D, degré de confiance après prise en compte des données

$$p(H|D) = p(H)\, p(D|H) / p(D)$$



http://imgs.xkcd.com/comics/seashell.png

Les statistiques Bayésiennes

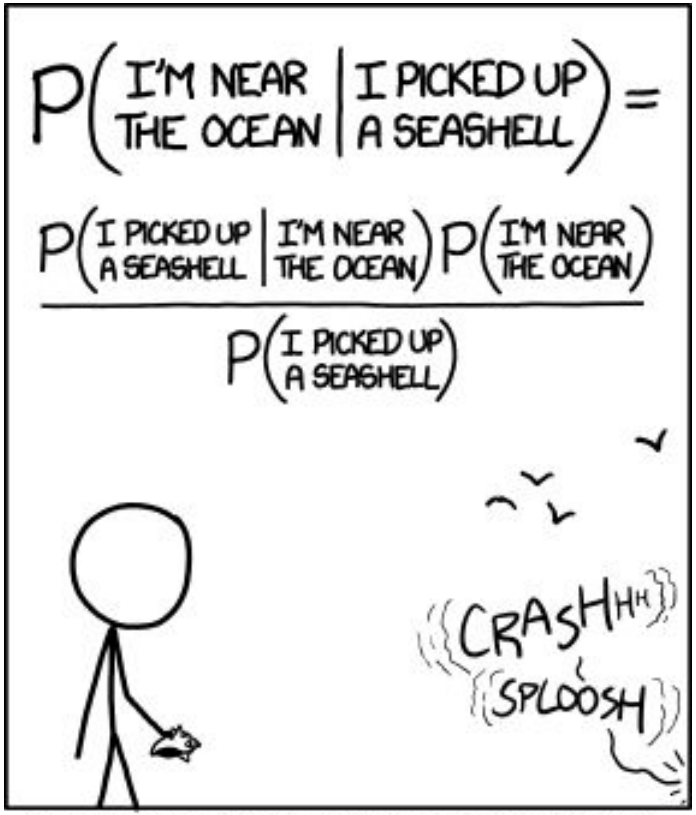
(en 1 diapo)

H=hypothèse et D=données

probabilité de l'hypothèse H connaissant les données D, degré de confiance après prise en compte des données

$$p(H \mid D) = p(H)\, p(D \mid H) / p(D)$$

A priori probabilité de l'hypothèse H avant d'avoir vu les données D, degré de confiance en l'hypothèse



http://imgs.xkcd.com/comics/seashell.png

Les statistiques Bayésiennes

(en 1 diapo)

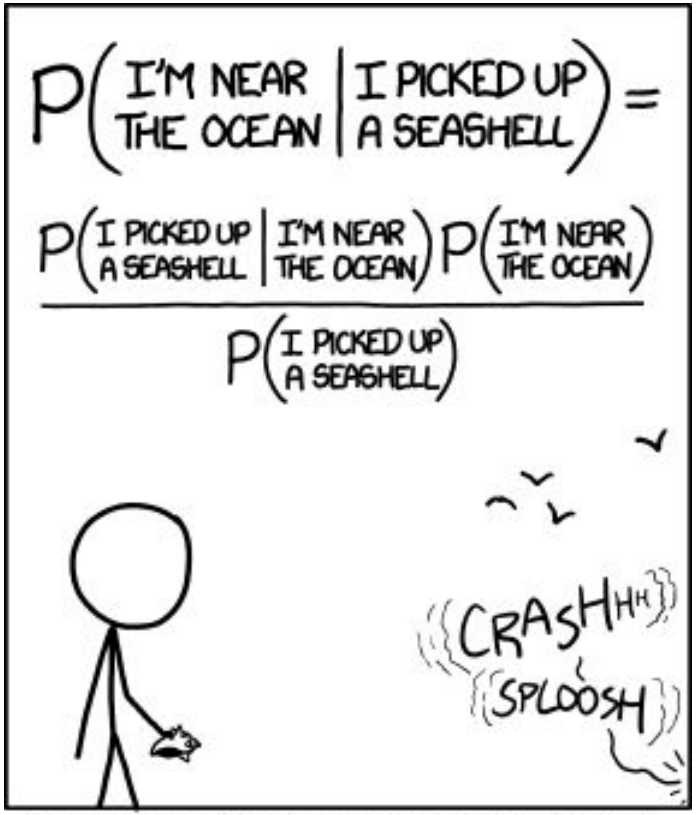
H=hypothèse et D=données

probabilité de l'hypothèse H connaissant les données D, degré de confiance après prise en compte des données

$$p(H \mid D) = p(H)\, p(D \mid H) / p(D)$$

A priori probabilité de l'hypothèse H avant d'avoir vu les données D, degré de confiance en l'hypothèse

Vraisemblance, degré de compatibilité entre hypothèse et données



http://imgs.xkcd.com/comics/seashell.png

Les statistiques Bayésiennes

(en 1 diapo)

H=hypothèse et D=données

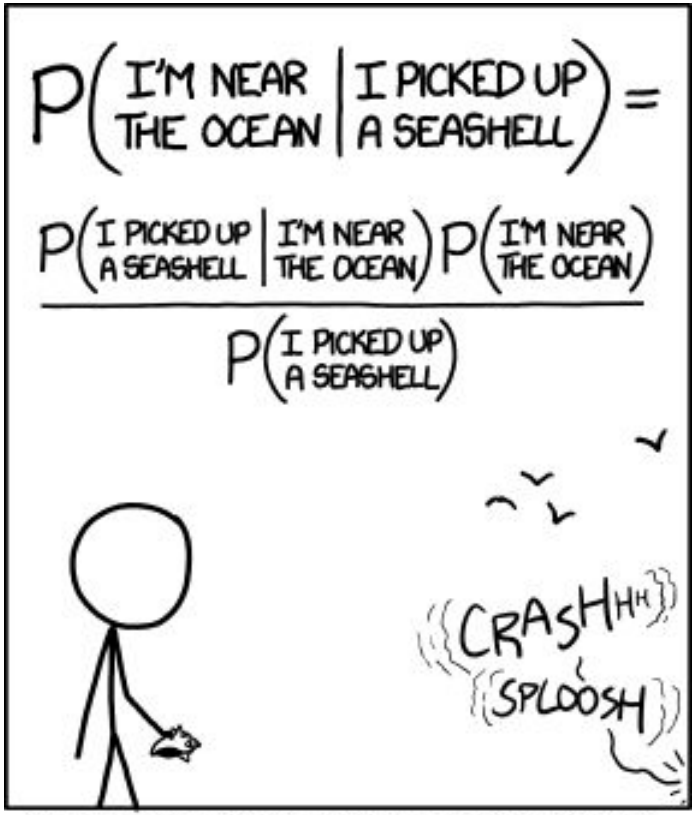
probabilité de l'hypothèse H connaissant les données D, degré de confiance après prise en compte des données

$$p(H \mid D) = p(H)\, p(D \mid H) / p(D)$$

A priori probabilité de l'hypothèse H avant d'avoir vu les données D, degré de confiance en l'hypothèse

Vraisemblance, degré de compatibilité entre hypothèse et données

Probabilité d'observation



http://imgs.xkcd.com/comics/seashell.png

Les statistiques Bayésiennes

(~~en 1 diapo~~, en fait 2)

Exemple (An introduction to Bayesian Statistics using Python https://www.slideshare.net/freshdatabos/an-introduction-to-bayesian-statistics-using-python)

Il y a 2 bols contenant des cookies. Bol 1 = 10 chocolat + 30 vanille, Bol 2 = 20 chocolat + 20 vanille
Fred prend un bol au hasard et un cookie au hasard. Le cookie est à la vanille.
Quelle est la probabilité que Fred ait pris le cookie dans le bol 1?

H : hypothèse que le cookie vienne du bol 1
D : le cookie est à la vanille

$$p(H|D) = p(H)\, p(D|H) / p(D)$$

Les statistiques Bayésiennes

(~~en 1 diapo~~, en fait 2)

Exemple (An introduction to Bayesian Statistics using Python https://www.slideshare.net/freshdatabos/an-introduction-to-bayesian-statistics-using-python)

Il y a 2 bols contenant des cookies. Bol 1 = 10 chocolat + 30 vanille, Bol 2 = 20 chocolat + 20 vanille
Fred prend un bol au hasard et un cookie au hasard. Le cookie est à la vanille.
Quelle est la probabilité que Fred ait pris le cookie dans le bol 1?

H : hypothèse que le cookie vienne du bol 1
D : le cookie est à la vanille

$$p(H|D) = p(H)\, p(D|H) / p(D)$$

p(H) = ½ (1 chance sur 2 d'avoir pris le bol 1)

Les statistiques Bayésiennes

(~~en 1 diapo~~, en fait 2)

Exemple (An introduction to Bayesian Statistics using Python https://www.slideshare.net/freshdatabos/an-introduction-to-bayesian-statistics-using-python)

Il y a 2 bols contenant des cookies. Bol 1 = 10 chocolat + 30 vanille, Bol 2 = 20 chocolat + 20 vanille
Fred prend un bol au hasard et un cookie au hasard. Le cookie est à la vanille.
Quelle est la probabilité que Fred ait pris le cookie dans le bol 1?

H : hypothèse que le cookie vienne du bol 1
D : le cookie est à la vanille

$$p(H|D) = p(H)\, p(D|H) / p(D)$$

p(H) = ½ (1 chance sur 2 d'avoir pris le bol 1)
p(D|H) = ¾ (si le cookie vient bien du bol 1, Fred avait 3 chances sur 4 d'avoir un cookie à la vanille)

Les statistiques Bayésiennes

(~~en 1 diapo~~, en fait 2)

Exemple (An introduction to Bayesian Statistics using Python https://www.slideshare.net/freshdatabos/an-introduction-to-bayesian-statistics-using-python)

Il y a 2 bols contenant des cookies. Bol 1 = 10 chocolat + 30 vanille, Bol 2 = 20 chocolat + 20 vanille
Fred prend un bol au hasard et un cookie au hasard. Le cookie est à la vanille.
Quelle est la probabilité que Fred ait pris le cookie dans le bol 1?

H : hypothèse que le cookie vienne du bol 1
D : le cookie est à la vanille

$$p(H|D) = p(H)\, p(D|H) / p(D)$$

p(H) = ½ (1 chance sur 2 d'avoir pris le bol 1)
p(D|H) = ¾ (si le cookie vient bien du bol 1, Fred avait 3 chances sur 4 d'avoir un cookie à la vanille)
p(D) = ⅝ (il y a 50 cookies à la vanille sur un total de 80 cookies)

Les statistiques Bayésiennes

(~~en 1 diapo~~, en fait 2)

Exemple (An introduction to Bayesian Statistics using Python https://www.slideshare.net/freshdatabos/an-introduction-to-bayesian-statistics-using-python)

Il y a 2 bols contenant des cookies. Bol 1 = 10 chocolat + 30 vanille, Bol 2 = 20 chocolat + 20 vanille
Fred prend un bol au hasard et un cookie au hasard. Le cookie est à la vanille.
Quelle est la probabilité que Fred ait pris le cookie dans le bol 1?

H : hypothèse que le cookie vienne du bol 1
D : le cookie est à la vanille

$$p(H|D) = p(H)\, p(D|H) / p(D)$$

p(H) = ½ (1 chance sur 2 d'avoir pris le bol 1)
p(D|H) = ¾ (si le cookie vient bien du bol 1, Fred avait 3 chances sur 4 d'avoir un cookie à la vanille)
p(D) = ⅝ (il y a 50 cookies à la vanille sur un total de 80 cookies)

p(H|D) = ⅗

Les statistiques Bayésiennes

(~~en 1 diapo~~, en fait 2)

Exemple (An introduction to Bayesian Statistics using Python https://www.slideshare.net/freshdatabos/an-introduction-to-bayesian-statistics-using-python)

Il y a 2 bols contenant des cookies. Bol 1 = 10 chocolat + 30 vanille, Bol 2 = 20 chocolat + 20 vanille
Fred prend un bol au hasard et un cookie au hasard. Le cookie est à la vanille.
Quelle est la probabilité que Fred ait pris le cookie dans le bol 1?

H : hypothèse que le cookie vienne du bol 1
D : le cookie est à la vanille

$$p(H|D) = p(H)\, p(D|H) / p(D)$$

A prior = 50%
A posteriori = 60%
on a augmenté notre degré de confiance en notre hypothèse

$p(H) = ½$ (1 chance sur 2 d'avoir pris le bol 1)
$p(D|H) = ¾$ (si le cookie vient bien du bol 1, Fred avait 3 chances sur 4 d'avoir un cookie à la vanille)
$p(D) = ⅝$ (il y a 50 cookies à la vanille sur un total de 80 cookies)

$p(H|D) = ⅗$

RJags

- JAGS Just Another Gibbs Sampler (Martyn Plummer et al.), clone de BUGS (Bayesian analysis Using Gibbs Sampling), similar to OpenBUGS or WinBUGs

- Avantage : interface R, plus simple que BUGS, fonctionne bien sur Ubuntu, beaucoup de tutoriels disponible

- Pré-requis : installation de R, installation de JAGS (http://mcmc-jags.sourceforge.net/) et des packages coda, rjags, R2jags

- JAGS existe depuis Décembre 2002, rjags depuis Mai 2008
  Version JAGS 4.3 (Juillet 2017), rjags 4.7 (Octobre 2018)

Principe du language JAGS

1) Définir un modèle en langage BUGS

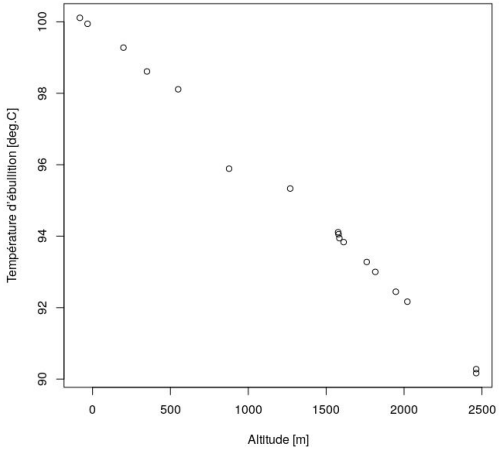
2) Lire ce modèle et créer un objet ‘jags’ dans R

3) Faire tourner ce modèle (burn-in, itérations)

4) Extraire les échantillons de ce modèle comme distribution a posteriori

Exemple simple - Régression linéaire

1) Jeu de données
2) Écriture modèle (2 façons : rjags ou R2jags)




```
########### rjags #######
model_rjags <- " model {
                # Likelihood
                for (i in 1:N) {
                    temp[i] ~ dnorm(mu[i] , tau)
                    mu[i] <- a + b*alt[i] # linear regression expression
                    }
                # Priors
                a ~ dnorm(0, 0.0001) # intercept
                b ~ dnorm(0, 0.0001) #slope
                sigma ~ dunif(0, 100) #standard deviation
                tau <- 1 / (sigma * sigma) #precision
                }"
```

```
########### R2jags #######
model_r2jags <- function() {
                # Likelihood
                for (i in 1:N) {
                    temp[i] ~ dnorm(mu[i] , tau)
                    mu[i] <- a + b*alt[i] # linear regression expression
                    }
                # Priors
                a ~ dnorm(0, 0.0001) # intercept
                b ~ dnorm(0, 0.0001) # slope
                sigma ~ dunif(0, 100) # standard deviation
                tau <- 1 / (sigma * sigma) # precision
                }
```

```
> ebullition
         alt      temp
1   2463.145  90.27778
2   2463.145  90.16667
3   2021.200  92.16667
4   1947.085  92.44444
5   1815.325  93.00000
6   1760.425  93.27778
7   1612.195  93.83333
8   1584.745  93.94444
9   1576.510  94.11111
10  1579.255  94.05556
11  1269.070  95.33333
12   876.535  95.88889
13   349.495  98.61111
14   549.880  98.11111
15   198.520  99.27778
16   -32.060  99.94444
17   -81.470 100.11111
```

Exemple simple - Régression linéaire

Option template.jags()

```
model_lm<-lm(temp~alt)
template.jags(model_lm,ebullition)
```

Écriture d'un fichier texte contenant modèle voulu en langage JAGS

```
############################################################################################################
############################################################################################################
#### JAGS model file written by runjags version 2.0.4-2 on 2019-02-13 09:55:24
############################################################################################################
############################################################################################################

### Model template as follows - ensure this is syntactically correct before running the model!

model{

# In the BUGS/JAGS language we must use an explicit for loop:
for(i in 1:N){
    # These lines describe the response distribution and linear model terms:
    temp[i] ~ dnorm(regression_fitted[i], regression_precision)
    regression_residual[i] <- temp[i] - regression_fitted[i]
    regression_fitted[i] <- intercept + alt_coefficient * alt[i]
}

# These lines give the prior distributions for the parameters to be estimated:
regression_precision ~ dgamma(0.001, 0.001)
intercept ~ dnorm(0, 10^-6)
alt_coefficient ~ dnorm(0, 10^-6)
resid.sum.sq <- sum(regression_residual^2)
}
```

Exemple simple - Régression linéaire

3) Initialisation des priors (non obligatoire) et des paramètres d'intérêt

```
init_values <- function(){ list(a = rnorm(1,0,1), b = rnorm(1,0,1), sigma = runif(1,0,3)) }
# not compulsory to initialise parameters, JAGS can do it automatically

params <- c("a", "b", "sigma")
# list of parameters of interest
```

4) Run modèle (R2jags)

```
fit_model_r2jags <- jags(   data = list('alt' = ebullition$alt,'temp' = ebullition$temp,'N' = nrow(ebullition)),
                    inits = init_values,
                    parameters.to.save = params,
                    model.file = model_r2jags,
                    n.chains = 4, # number of Markov chains
                    n.iter = 10000, # number of iteration per chain
                    n.burnin = 1000, # number of iteration to discard at beginning
                    n.thin = 10) # thinning rate, if 10: return 1 every 10 samples

fit_model_r2jags<-autojags(fit_model_r2jags) # function to automatically update model until it converges
```

Exemple simple - Régression linéaire

5) Résultats

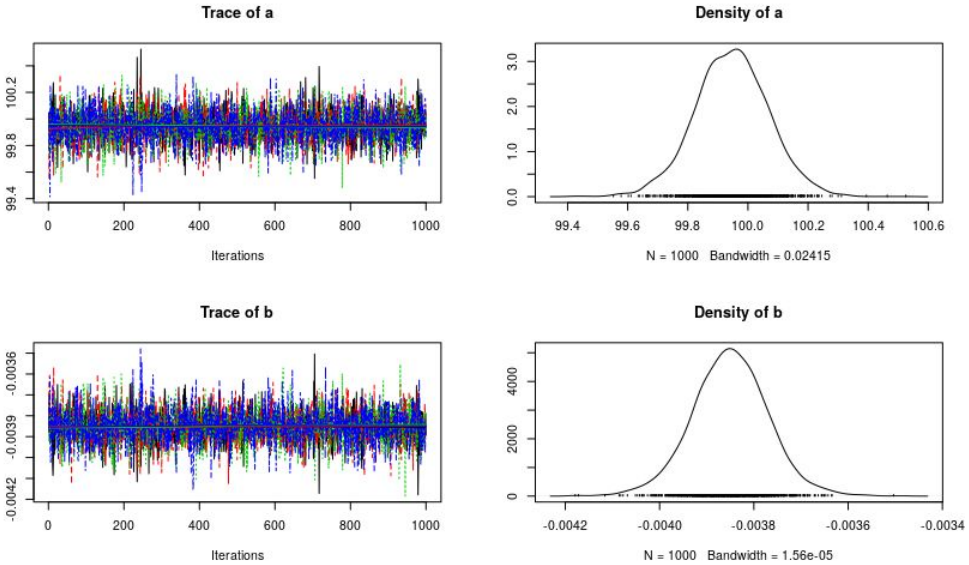
```
> fit_model_r2jags
Inference for Bugs model at "/tmp/Rtmpk2FEeh/model13c21f9581ce.txt", fit using jags,
 4 chains, each with 10000 iterations (first 1000 discarded), n.thin = 10
 n.sims = 3600 iterations saved
         mu.vect sd.vect   2.5%    25%    50%     75%   97.5%  Rhat n.eff
a         99.944   0.125 99.697 99.864 99.943 100.023 100.193 1.001  3600
b         -0.004   0.000 -0.004 -0.004 -0.004  -0.004  -0.004 1.000     1
sigma      0.270   0.054  0.185  0.231  0.263   0.299   0.394 1.001  3500
deviance   1.982   2.805 -1.236 -0.039  1.333   3.239   9.391 1.003  1400

For each parameter, n.eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor (at convergence, Rhat=1).

DIC info (using the rule, pD = var(deviance)/2)
pD = 3.9 and DIC = 5.9
DIC is an estimate of expected predictive error (lower deviance is better).
```

```
> summary(model_lm)

Call:
lm(formula = temp ~ alt)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6816 -0.1232  0.0429  0.1094  0.2833

Coefficients:
              Estimate  Std. Error t value            Pr(>|t|)
(Intercept) 99.94425440  0.11317140  883.12 <0.0000000000000002 ***
alt         -0.00384899  0.00007439  -51.74 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2467 on 15 degrees of freedom
Multiple R-squared:  0.9944,    Adjusted R-squared:  0.9941
F-statistic:  2677 on 1 and 15 DF,  p-value: < 0.00000000000000022
```

Exemple simple - Régression linéaire

5) MCMC simulations

```
lm_mcmc_r2jags<- as.mcmc(fit_model_r2jags)
plot(lm_mcmc_r2jags)
```



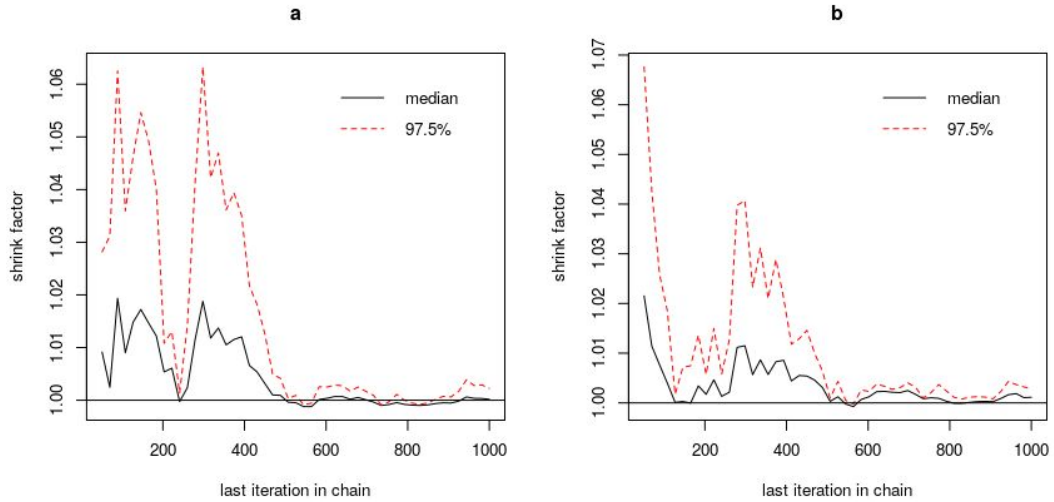
Trace : trajectoire des chaînes MCMC au cours des itérations

Distributions de a et b sont Normales (comme attendu)

Exemple simple - Régression linéaire

6) Diagnostiques

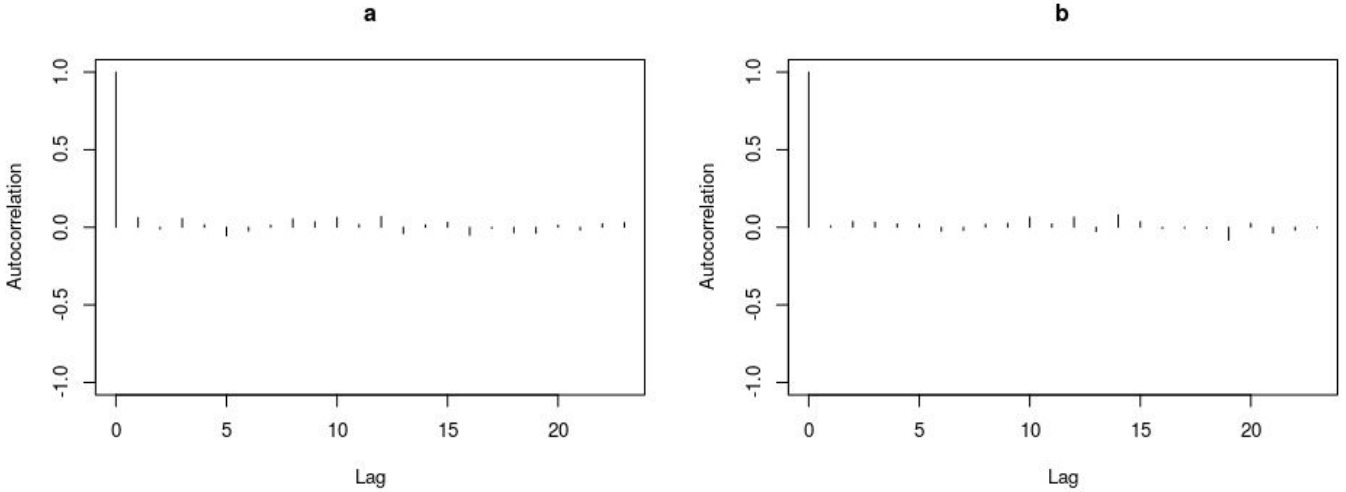
- Gelman : convergence gelman.diag(), gelman.plot()



Exemple simple - Régression linéaire

6) Diagnostiques

- Autocorrelation : paramétrisation, corrélation entre échantillons, temps jusqu'à convergence autocorr.diag(), autocorr.plot()



Exemple simple - Régression linéaire

6) Diagnostiques

- Taille effective d'échantillon : taille d'échantillon ajustée pour l'autocorrélation effectiveSize()

a	b	deviance	sigma
3616.107	3600.000	3600.000	3299.465

Taille d'échantillon = n.chains * (n.iter+(update)-n.burnin) / n.thin

= 4 * (10000-1000) / 10 = 3600

Plus de possibilitées

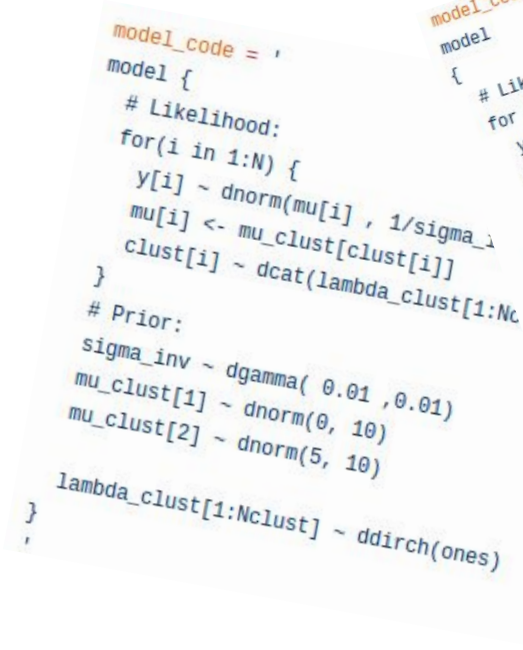
- Panel de distributions

- Modèles mixtes, hiérarchiques et plus complexes

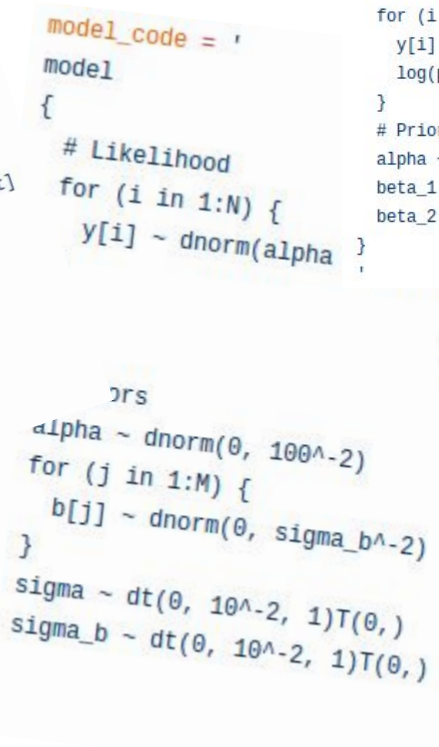
- Prendre en compte des priors informatifs

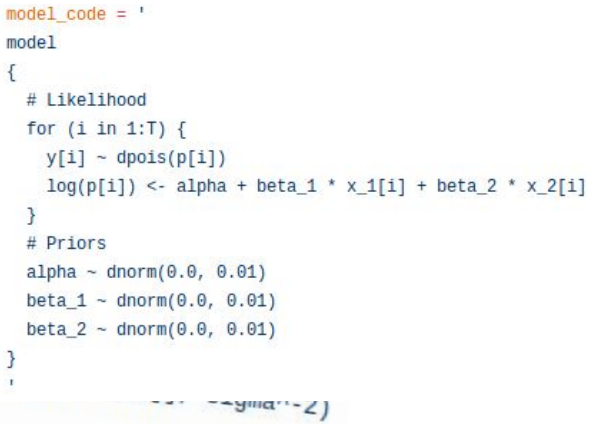
De nombreux exemples sur GitHub Andrew Parnell (https://github.com/andrewcparnell/jags_examples)

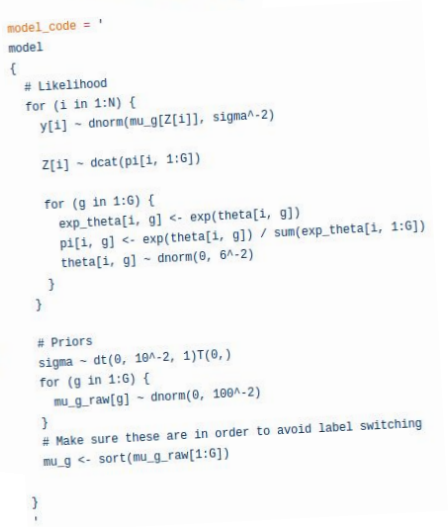
Merci !


model_code = '
model {
# Likelihood:
for(i in 1:N) {
y[i] ~ dnorm(mu[i] , 1/sigma_
mu[i] <- mu_clust[clust[i]]
clust[i] ~ dcat(lambda_clust[1:Nc
}
# Prior:
sigma_inv ~ dgamma(0.01 ,0.01)
mu_clust[1] ~ dnorm(0, 10)
mu_clust[2] ~ dnorm(5, 10)
lambda_clust[1:Nclust] ~ ddirch(ones)
}
'


model_code = '
model
{
# Likelihood
for (t in 1:T) {
y[t] ~ dbeta(a[t], b[t])
a[t] <- mu[t] * phi
b[t] <- (1 - mu[t]) * phi
logit(mu[t]) <- alpha + beta * x[t]
}
# Priors
alpha ~ dnorm(0, 10^-2)
beta ~ dnorm(0, 10^-2)
phi ~ dunif(0, 10)
}
'
odel_code = '
model
{
# Likelihood
for (i in 1:T) {
mu[i] = exp(beta_1 * x_1[
t[i] ~ dexp(mu[i] * lambd
}
# Priors
lambda_0 ~ dgamma(1, 1)
beta_1 ~ dnorm(0.0, 0.01)
beta_2 ~ dnorm(0.0, 0.01)
}
'


model_code = '
model
{
# Likelihood
for (i in 1:N) {
y[i] ~ dnorm(alpha
ors
alpha ~ dnorm(0, 100^-2)
for (j in 1:M) {
b[j] ~ dnorm(0, sigma_b^-2)
}
sigma ~ dt(0, 10^-2, 1)T(0,)
sigma_b ~ dt(0, 10^-2, 1)T(0,)
}
'


model_code = '
model
{
# Likelihood
for (i in 1:T) {
y[i] ~ dpois(p[i])
log(p[i]) <- alpha + beta_1 * x_1[i] + beta_2 * x_2[i]
}
# Priors
alpha ~ dnorm(0.0, 0.01)
beta_1 ~ dnorm(0.0, 0.01)
beta_2 ~ dnorm(0.0, 0.01)
}
'


model_code = '
model
{
# Likelihood
for (i in 1:N) {
y[i] ~ dnorm(mu_g[Z[i]], sigma^-2)
Z[i] ~ dcat(pi[i, 1:G])
for (g in 1:G) {
exp_theta[i, g] <- exp(theta[i, g])
pi[i, g] <- exp(theta[i, g]) / sum(exp_theta[i, 1:G])
theta[i, g] ~ dnorm(0, 6^-2)
}
}
# Priors
sigma ~ dt(0, 10^-2, 1)T(0,)
for (g in 1:G) {
mu_g_raw[g] ~ dnorm(0, 100^-2)
}
# Make sure these are in order to avoid label switching
mu_g <- sort(mu_g_raw[1:G])
}
'

Exemple simple - Régression linéaire

4) Run modéle (rjags)

```
fit_model_rjags <- jags.model(  textConnection(model_rjags), # or link to text file containing model
                              data = list('alt' = ebullition$alt,'temp' = ebullition$temp,'N' = nrow(ebullition)),
                              inits=init_values,
                              n.chains = 4,
                              n.adapt = 1000) # retuning/adaptation of parameters using x samples from posterior

update(fit_model_rjags , 1000) # burn-in period

samps.jags <- jags.samples(fit_model_rjags,
             params,
             n.iter = 10000,
             thin = 10)

samps.coda <- coda.samples(fit_model_rjags,
             params,
             n.iter = 10000,
             thin = 10)
```